Steven M. Hoffberg

From: Steven M. Hoffberg [steve@hoffberg.org]

Sent: Thursday, November 18, 2004 12:14 PM

To: 'Nguyen, Nga'

Subject: 09/599,163, Oliver et al.

Attachments: db_access.c.txt; hash.c.txt; http_clickshare.c.txt; process_log.txt; profile.c.txt; service_db.c.txt; test_client.c.txt; token.c.txt; tvs_client.c.txt; user_db.c.txt

Db_access.c

```
/* ---------------------------------------------------------------------
 * ---------------------------------------------------------------------
 * access routines for master Clickshare transaction logging database(s)
 * Copyright (c) 1995 Newshare Corporation
 * ---------------------------------------------------------------------
 * ---------------------------------------------------------------------
 */

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/time.h>
#include <sys/un.h>
#include <time.h>
#include <errno.h>
#include <math.h>

#include "log_db.h"
#include "log.h"                        /* syslog interface */
#include "msql.h"

#define PUBLIC
#define PRIVATE static

PRIVATE char query[2048];

PRIVATE int num_results = 0;  /* number of result records from prior query */

PRIVATE char *months[12] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun",\
                            "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};

/* ---------------------------------------------------------------------
 * xlate_month - convert a number month to a char string
 * ---------------------------------------------------------------------
 */
```

```
PUBLIC char *
xlate_month(int mon)
{
  if ((mon < 1) || (mon > 12)) return (char *) NULL;

  return months[(mon - 1)];
}

/* ---------------------------------------------------------------------
 * db_num_results:  return the number of results from last DB SELECT query
 * ---------------------------------------------------------------------
 */

PUBLIC int
db_num_results()
{
  return num_results;
}

/* ---------------------------------------------------------------------
 * db_set_num_results: set the number of results from last DB SELECT query
 * ---------------------------------------------------------------------
 */

PUBLIC void
db_set_num_results(int num)
{
  num_results = num;
}


/* ---------------------------------------------------------------------
 * db_load_record
 * NOTE: SUPER MEGA ULTRA "field order dependent".  See log_db.h
 * #define ALL_LOG_FIELDS for order
 * ---------------------------------------------------------------------
 */

PRIVATE LOG_RECORD
db_load_record(m_row row)
{
  unsigned long number;
  LOG_RECORD rec;

  rec = create_log_record();
  if (!rec) {
    sprintf(msgString, "db_load_record: out of memory");
    LogMsg(LOG_ERR, msgString);
    return (LOG_RECORD) NULL;
  }
```

```
  sscanf(row[0], "%ld", &number);
  rec->rec_id  = number;
  sscanf(row[1], "%ld", &number);
  rec->user_id  = number;

  strcpy(rec->host_string, row[2]);
  strcpy(rec->date, row[3]);

  sscanf(row[4], "%ld", &number);
  rec->session_id    = number;
  sscanf(row[5], "%ld", &number);
  rec->service_class = number;

  strcpy(rec->flags, row[6]);

  sscanf(row[7], "%ld", &number);
  rec->homepm_id = number;
  sscanf(row[8], "%ld", &number);
  rec->page_class = number;
  sscanf(row[9], "%ld", &number);
  rec->contentpm_id = number;

  strcpy(rec->method, row[10]);
  strcpy(rec->url, row[11]);

  sscanf(row[12], "%ld", &number);
  rec->nbytes = number;

  return rec;
}

/* ---------------------------------------------------------------------
 * db_insert:  insert a tranaction record into the master logging database
 * ---------------------------------------------------------------------
 */

PUBLIC int
db_insert(DB_SOCKET db_sock, LOG_RECORD rec)
{
  int sock = (int) db_sock;

  if (rec == (LOG_RECORD) NULL) return 0;

  /* make sure my unique key is here, else I'm dead */

  if (rec->rec_id == 0) return 0;

  /* need to be sure field order matches record order as I am not specifyning
   * fields.
   */
```

```
  sprintf(query,
          "INSERT INTO %s VALUES (%ld, %ld, '%s', '%s', %ld, %ld, '%s', %ld, %ld, %ld, '%s',
'%s', %d)",
          LOG_TABLE_NAME,
          rec->rec_id,
          rec->user_id,
          rec->host_string,
          rec->date,
          rec->session_id,
          rec->service_class,
          rec->flags,
          rec->homepm_id,
          rec->page_class,
          rec->contentpm_id,
          rec->method,
          rec->url,
          rec->nbytes);
  if (msqlQuery(sock, query) < 0) {
    sprintf(msgString, "db_insert: %s\n", msqlErrMsg);
    LogMsg(LOG_ERR, msgString);
    return -1;
  }
  return 0;
}

/* ----------------------------------------------------------------------
 * db_retrieve: pull entries from a specific query out of the
 * database.
 * ----------------------------------------------------------------------
 */

PUBLIC LOG_RESULT
db_retrieve(DB_SOCKET db_sock)
{
  int n;
  LOG_RESULT results;
  LOG_RECORD rec;

  m_result *res   = (m_result *) NULL;
  m_row     row   = (m_row) NULL;

  /* store the result and obtain it */

  res = msqlStoreResult();
  if (!res || (msqlNumRows(res) < 1)) {
    fprintf(stderr,
            "db_retrieve: no entries found to match query\n");
    db_set_num_results(-1);
    return (LOG_RESULT) NULL;
  }
```

```
/* save the number of records (which at this point we know is > 1) */

db_set_num_results(msqlNumRows(res));

/* create the array to store the results in */

results = (LOG_RESULT) malloc(db_num_results() * sizeof(LOG_RECORD));
if (!rec) {
  msqlFreeResult(res);
  sprintf(msgString, "db_retrieve: out of memory\n");
  LogMsg(LOG_ERR, msgString);
  db_set_num_results(-1);
  return (LOG_RESULT) NULL;
}

/* create all records */

for (n = 0; n < db_num_results(); n++) {

  msqlDataSeek(res, n);      /* set position in row output array from res */

  row = msqlFetchRow(res);
  if (!row) {
    msqlFreeResult(res);
    db_free_result(results, n-1);
    sprintf(msgString, "db_retrieve: error fetching entry\n");
    LogMsg(LOG_ERR, msgString);
    return (LOG_RESULT) NULL;
  }

  /* create a place to store the results */

  rec = create_log_record();
  if (!rec) {
    msqlFreeResult(res);
    db_free_result(results, n-1);
    sprintf(msgString, "db_retrieve: out of memory\n");
    LogMsg(LOG_ERR, msgString);
    return (LOG_RESULT) NULL;
  }

  /* load the query results into a log record ... */

  if ((rec = db_load_record(row)) < 0) {
    msqlFreeResult(res);
    db_free_result(results, n-1);
    sprintf(msgString, "db_retrieve: error loading up results\n");
    LogMsg(LOG_ERR, msgString);
    return (LOG_RESULT) NULL;
  }
```

```
    else {
      results[n] = rec;          /* ... and store the in results array */
    }
  }

  msqlFreeResult(res);
  return results;
}

/* ------------------------------------------------------------------------
 * db_retrieve_session: pull entries from a specific session out of the
 * database.
 * ------------------------------------------------------------------------
 */

PUBLIC LOG_RESULT
db_retrieve_session(DB_SOCKET db_sock, unsigned long session_id)
{
  int sock = (int) db_sock;

  LOG_RESULT results;

  /* construct the database query */

  sprintf(query, "SELECT %s FROM %s WHERE session_id=%ld",
          ALL_LOG_FIELDS, LOG_TABLE_NAME, session_id);

  /* do it */

  if (msqlQuery(sock, query) < 0) {
    fprintf(stderr, "db_retrieve_session: %s\n", msqlErrMsg);
    return (LOG_RESULT) NULL;
  }

  results = db_retrieve(db_sock);
  return results;
}

/* ------------------------------------------------------------------------
 * db_retrieve_date: pull entries from a specific user/date out of the database
 * ------------------------------------------------------------------------
 */

PUBLIC LOG_RESULT
db_retrieve_date(DB_SOCKET db_sock, unsigned long u_id, int mon, int day, int yr)
{
  int sock = (int) db_sock;

  LOG_RESULT results;

  /* construct the database query */
```

```
  sprintf(query, "SELECT %s FROM %s WHERE user_id=%ld AND date LIKE '%%%d/%s/%
d%%'",
        ALL_LOG_FIELDS, LOG_TABLE_NAME, u_id, day, xlate_month(mon), yr);

  /* do it */

  if (msqlQuery(sock, query) < 0) {
    fprintf(stderr, "db_retrieve_date: %s\n", msqlErrMsg);
    return (LOG_RESULT) NULL;
  }

  results = db_retrieve(db_sock);
  return results;
}
```

Very truly yours,

Steven M. Hoffberg
Milde & Hoffberg, LLP
Suite 460
10 Bank Street
White Plains, NY 10606
(914) 949-3100 tel.
(914) 949-3416 fax
steve@hoffberg.org
www.hoffberg.org